

**LINK***group*

Distance Learning System



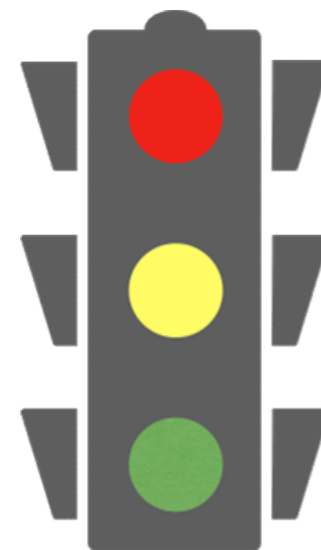
# Strukture za kontrolu toka

Python and programming fundamentals

# Kontrola toka

---

- Kontrola toka je set instrukcija, naredbi koje omogućavaju da program donosi neke odluke
- Kontrola toka bazira se na proveru toga da li je neki uslov ispunjen i u zavisnosti od toga se odlučuje na koji način će se nastaviti izvršavanje programa
- Najčešća upotreba kontrole toka je za to da u samom toku programa mi postavimo neko pitanje i na osnovu odgovora koji se dobija na to pitanje program nastavlja da se obavlja na jedan ili na drugi način
- Kada stojimo na semaforu, ukoliko je zeleno svetlo za pešake, preći ćemo ulicu, odnosno određeni uslov je ispunjen i mi smo preduzeli određenu akciju, a ukoliko je crveno, sačekaćemo



# If struktura

---

- Najpoznatija struktura za kontrolu toka u programiranju je **if** struktura
- If podrazumeva uslovno izvršavanje određenog bloka koda

- If struktura započinje ključnom rečju **if(expression):** kojoj sledi **block**

izraz u malim zagradama ili bez njih, dvotačka, a

# If struktura

---

- Izraz u if strukturi mora rezultirati boolean vrednošću
- Čak i ako tip koji je prosleđen u izrazu nije boolean, on će biti

Blok koda započinje  
implicitno konvertovan u taj tip, pod oc  
uvlakom neodređene širine  
i obuhvata svaku sledeću  
liniju sa istom uvlakom

```
x = True
if(x):
    print("Hello")
```

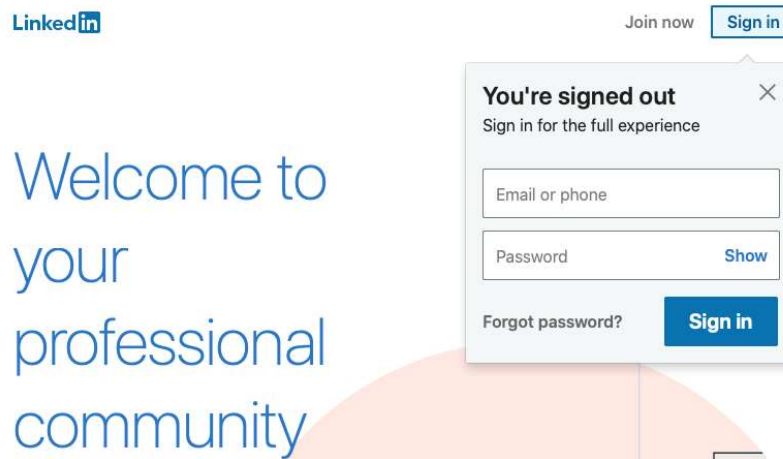
- Program će na izlazu odštampati poruku **Hello**

# Gde sve koristimo if strukturu

**If** struktura je okosnica programiranja jer je to primarni način da program ne bude linearan. Gotovo da ne postoji program u kome se ne koristi

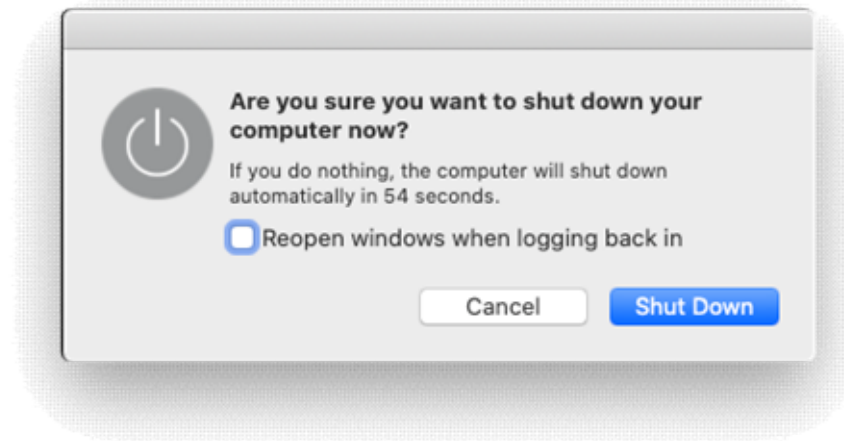
**If** struktura nam omogućava da "udahnemo život" boolean rezultatima dobijenim različitim izrazima

```
if(userLoggedIn):  
    showSignedOut()
```



# Gde sve koristimo if strukturu

---



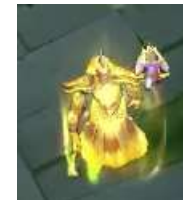
```
if(pressedButton == "Shut Down"):
    if(reopenCheckbox == True):
        saveState()
        shutDown()
```

# Gde sve koristimo if strukturu

---



```
if(pressedButton == ultiButton  
and heroMana >= ultiMana):  
    fireMagic(4)
```



```
if(zhonyaActivated == True and  
heroMana >= zhonyaMana):  
    useZhonya()
```

# Ugnježdjeni if uslovi

---

- Jedan if uslov može usloviti druge uslove

- Tada se uslovljavanje može nastaviti u novim uvlakama

```
ultiPressed = True
mana        = 100
manaCost    = 80
heroAlive   = True

if(ultiPressed):
    if(heroAlive):
        if(mana>=manaCost):
            print("Ulti fired!!!")
```




# Vežba 1 (ppf-ex05 age.py)

---

Na početku programa korisnik unosi broj godina. Ukoliko je broj godina veći ili

```
usage = int(input('Your age? '))  
if(usage>=13):  
    print("Access allowed")
```



```
Your age? 13  
Access allowed
```

# Vežba 2 (ppf-ex05 menu.py)

- Stavke menija definisane su brojevima:  
1 = prikaz proizvoda, 2 = kupovina proizvoda, 3 = izlaz iz programa
- Proizvod u programu definisan je promenljivima:  
productName, productPrice...
- Korisnik unosi kolicinu novca u aplikaciju, a zatim startuje neku od stavki iz menija
- Program izvršava operaciju koja odgovara odabranoj stavki

```
Enter balance: 150
Enter command: 3
Good bye
```

```
menuShowProduct = 1
menuBuyProduct  = 2
menuExitProgram = 3
productName     = "iPhone"
productPrice    = 180
userBalance     = int(input("Enter balance: "))
userCommand     = int(input("Enter command: "))
```

```
if(userCommand==1):
    print("Name: ", productName)
    print("Price: ", productPrice)
if(userCommand==2):
    if(userBalance>=productPrice):
        print("You bought the product")
    if(userBalance<productPrice):
        print("You don't have enough balance")
if(userCommand==3):
    print("Good bye")
```

**Strukture za kontrolu toka**

# Problem

---

- Ukoliko koristimo if strukturu u osnovnoj varijanti moguć je sledeći scenario:

```
keyPressed = True
if(keyPressed):
    print("Key is pressed")
    keyPressed = False
if(not keyPressed):
    print("Key is not pressed")
```



```
Key is pressed
Key is not pressed
```

- Primećujemo da je prva obrada slučaja, izmenila kontrolnu promenljivu, pa je došlo do aktivacije drugog slučaja, što nije dobro

# Rešenje - else izjava

- Izjava **else** omogućava izvršavanje bloka samo pod uslovom da prethodni blok nije izvršen
- Izjava **else**, ne koristi se nikada samostalno, već uvek unutar **if** strukture, u kombinaciji sa naredbom **if**

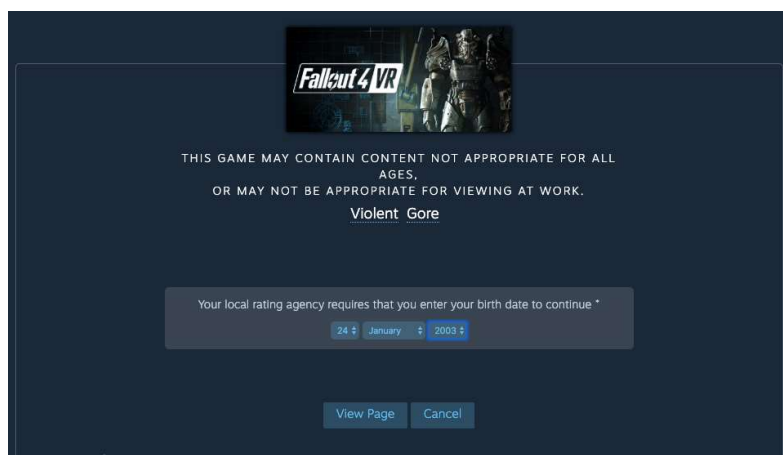
Ovaj blok će  
biti izvršen  
isključivo ako  
prvi blok ne  
bude izvršen

```
keyPressed = True
if(keyPressed):
    print("Key is pressed")
    keyPressed = False
else:
    print("Key is not pressed")
```

Key is pressed

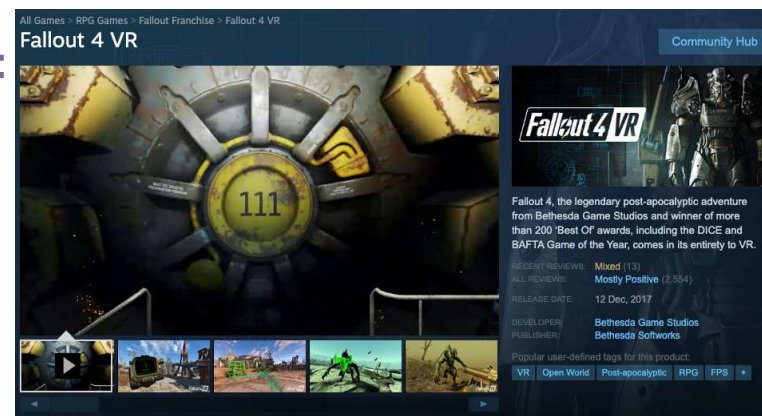
# Gde srećemo if else strukturu

- If i else se vrlo često sreću u kombinaciji
- Gotovo da nema slučaja koji se obrađuje a da nema alternativnu opciju



if(age >= 18):

else:

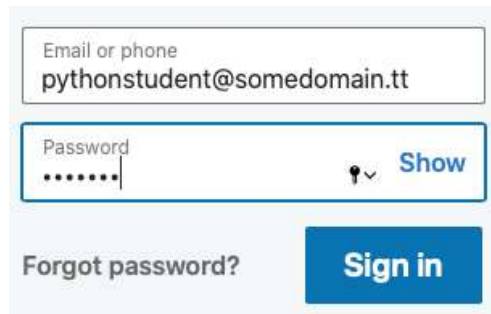


SORRY, BUT YOU'RE NOT PERMITTED TO VIEW THESE MATERIALS AT THIS TIME.


Home Retry

LINKgroup

# Gde srećemo if else strukturu



Email or phone  
pythonstudent@somedomain.tt

Password  
.....  Show

Forgot password? **Sign in**

`if(password == dbpassword):`

Authorized page

`else:`

Login error page

# Problem

---

- Sledeći primer je funkcionalan, ali postoje ugnježdene if strukture iako nam treba samo jedan slučaj

```
temperature = 20
if(temperature>10):
    if(temperature>20):
        print("You can use only shirt")
    else:
        print("You need jacket")
else:
    print("You will need jacket")
```

# Rešenje - elif izjava

---

- Elif se može posmatrati kao else sa dodatnim uslovom

- Ako je tačno, ništa se više ne proverava

- Ako ni jedan prethodni uslov nije ispunjen

```
temperature = 25
if(temperature > 20):
    print("You can use only shirt")
elif(temperature>10):
    print("You should use pullover")
else:
    print("You will need jacket")
```



## Vežba 3 (ppf-ex05 highscore.py)

---

- Potrebno je napraviti program koji od generiše slučajni ceo broj i od korisnika preuzima celobrojnu vrednost
- Ukoliko je broj koji je korisnik uneo, veći od broja dobijenog slučajnim izborom prikazuje se poruka da je pobeđen najbolji rezultat, dok se u suprotnom ispisuje poruka da treba još vežbati

```
Enter your score? 56  
Congrats! You beat the high score 49  
Highscore: 56
```

## Vežba 4 (ppf-ex05 moreless.py)

---

- Program preuzima od korisnika ceo broj. Potrebno je na izlazu prikazati poruku da je broj pozitivan, negativan ili nula, u zavisnosti od broja

```
Enter number? 0  
Number is zero
```

## Vežba 5 (ppf-ex05 oddeven.py)

---

- Program preuzima od korisnika ceo broj. Potrebno je na izlazu prikazati poruku da je broj paran ili neparan, u zavisnosti od broja

```
Enter number? 7  
Number is odd
```

## Vežba 6 (ppf-ex05 grade.py)

---

- Program preuzima od korisnika ocenu na osnovu koje ispisuje odgovarajuću poruku (1 nedovoljan, 2 dovoljan...)

```
Grade? 3  
Average grade
```

# Vežba 7 (ppf-ex05 quiz.py)

---

- Program postavlja tri pitanja korisniku. Korisnik odgovara u tekstualnom formatu i za svaki tačan odgovor dobija poene
- Nakon sva tri pitanja, prikazuju se tačni odgovori i broj osvojenih poena
- Pitanja formatirati po sopstvenom izboru

```
Best computer ever? amiga
Best game ever? fallout
Best film ever? matrix
##### RESULTS #####
Question 1: Best computer ever?
Answer: amiga
Question 2: Best game ever?
Answer: fallout
Question 3: Best film ever?
Answer: matrix
Your earned points: 3
```

# Vežba 8 (ppf-ex05 localize.py)

---

- Program od korisnika traži da unese željeni jezik
- Nakon unosa jezika, program prikazuje poruku na odabranom jeziku

```
Enter language? de  
Hallo
```

# Vežba 9 (ppf-ex05 odds.py)

- Program za pravljenje kvota, od korisnika traži da unese kvote za utakmicu iz liste kroz četiri pitanja
  - naziv utakmice (npr. Manchester United - Leeds)
  - tip 1 (domaći)
  - tip 0 (nerešeno)
  - tip 2 (gosti)
- Ukoliko korisnik ne unese ispravnu vrednost za kvotu program prekida izvršavanje
- Ukoliko je kvota manja ili jednaka vrednosti 1, program prekida izvršavanje
- Na kraju, program potvrđuje uspešan unos kvota i štampa unete podatke na izlazu

```
Enter match? Westham - Chelsea
Enter tip for home: abc
Sorry home tip is not numeric
```

## Napomena:

- Za forsirano napuštanje programa može se upotrebiti komanda: **exit(0)**
- Sledećom komandom može se proveriti da li je string broj:  
**"123.45".replace(".", "").isnumeric()**

```
Enter match? West Ham United - Chelsea
Enter tip for home: 1.2
Enter tip for draw: 1.5
Enter tip for away: 2.4
Thank you, your odd is:
Match: West Ham United - Chelsea
Home: 1.2
Draw: 1.5
Away: 2.4
```

# Vežba 10 (ppf-ex05 calculator.py)

---

- Potrebno je napraviti program kalkulator, koji radi sa četiri osnovne računske operacije
- Program od korisnika traži unos dva broja, a zatim željenu operaciju
- Nakon unosa brojeva i odabira operacije, program štampa rezultat na izlaz

```
Enter first operand? 4
Enter second operand? 5
Choose operation (add,sub,mul,div): mul
Result is: 20
```



# Vežba 11 (ppf-ex05 lcalculator.py)

---

- Prepraviti program iz prethodne vežbe tako da na početku traži od korisnika da unese jezik, a zatim sve poruke u programu prikazuje na odabranom jeziku:

```
Choose language (sr,en,de): de  
Geben Sie die erste Nummer ein? 12  
Geben Sie die zweite Nummer ein? 32  
Operation wählen (add,sub,mul,div): mul  
Das Ergebnis ist: 384
```

# Ternarni operator

---

- Specijalan zapis if else izraza u kome oba slučaja rezultiraju kontekstualno istom vrednošću

```
gender = "f"
if(gender=="m"):
    message = "Hello mister"
else:
    message = "Hey miss"
print(message)
```

- U ovakvim situacijama može se upotrebiti **ternarni operator**

# Ternarni operator

---

- Ternarni operator ima tri elementa. Vrednost u slučaju ispunjenog uslova, vrednost u slučaju neispunjenog uslova i sam uslov

```
message = "Hello mister" if gender=="m" else "Hey miss"
```



Positive outcome value **if** condition **else** negative outcome value